REMARKS

Claim 1 has been amended to include the subject matter of claim 6. It is respectfully submitted that the cited reference to Koenck does not teach bypassing a first processor if the first processor fails to respond. Cited in support of the contrary proposition is column 20, lines 53-54. This material talks about deactivating an application processor 48. It indicates that it could be deactivated by a complete electrical shutdown of the device. In such case, nothing is happening and there is no bypassing of the first processor if the whole device is turned off. It goes on to discuss removing the clocking signal from the application processor. But there is no indication that this is done in response to the processor failing to respond.

It is clear that the deactivation by a complete electrical shutdown of the device must be referring to the overall device, not the application processor 48, because it goes on to talk about deactivating the application processor 48 in response to a low current draw of the application processor. As pointed out in column 20, lines 43-47, the applications processor uses most of the power. Thus, in response to a power problem, it is apparently the goal to shutdown the applications processor because it consumes more power. But there is no suggestion of bypassing the first processor if the first processor fails to respond. Moreover, there is no suggestion that signals that would have gone to the first processor now go to the second processor.

Also cited is column 26, line 35 through column 27, line 4. But, again, there is nothing here that talks about bypassing the first processor when it fails to respond by diverting signals from the first to the second processor. For example, at column 26, lines 62-66, there is a discussion about when the application processor has completed its task, it communicates to the control processor 49 that its instructions have been executed and is shutdown to await further activation. Thus, there is no failure to respond and, in fact, the application processor announces its intention to shutdown. Moreover, there is nothing that is diverting from the first to the second processor.

Likewise, the material at column 27, lines 5-18, does not support the rejection. For example, at column 27, lines 12 and 13, there is a discussion of an impending shutdown due to a low battery indication. The text indicates that the application processor completes any operation if the low battery indication is still within tolerable limits or may suspend further data processing. But there is no bypassing of the first processor that fails to respond. This is simply a discussion that the first processor would complete all its tasks before it shuts down. There is no discussion

that it fails to respond and there is no discussion of any diverting signals from a first to a second processor.

Therefore, reconsideration is respectfully requested.

Claim 11 calls for establishing communications between an input/output device and a first processor to execute a first task. In response to the detection of an event, the communications from the input/output device are provided to a second processor that executes the first task in place of the first processor.

It is suggested in the office action that, in response to the detection of an event, the communications are provided to the second processor. An example is given where the event may be an incoming call, the user presses the call button 5A which, in turn, suspends the controller for the pocket game 28 and allows the caller information to be displayed on the first display. But there is no diverting of communications between the input/output device and the controller for the pocket game to another controller. The other controller is still doing what it is always doing, which is displaying information about the call. All that happens in the cited reference is that the pocket game controller is suspended. But none of its functions in the form of communications between itself and an input/output device are provided to a second processor in its stead. Thus, in the example given in the office action, the communications related to the caller information were never going to the controller for the pocket game because they would have no relation to the pocket game. Thus, there is none of the claimed changing of communications between one processor to another processor in response to the detection of an event.

As explained in column 6, lines 13-20, cited in paragraph 12 of the office action, the controller 22 issues the suspend to the controller for the pocket game and displays caller information to the first display (that was previously displaying the game in progress). In other words, there is no diversion of any communication between the input/output device in the first processor to a second processor. Different communications that would always have been handled by the second processor continue to be handled by the second processor.

The claim has been amended to clarify this difference in operation. The claim has been amended to call for the communications to execute a first task in the first processor and then, in response to the detection of an event, to provide the communications instead to a second processor so that the second processor (instead of the first processor) executes the first task.

In the cited reference, all that happens is each processor does what functions it always does. However, the control of a display is ceded from the controller for the pocket game to the controller 22. But a second processor does not execute a task that was previously executed by a previous processor. Each processor executes the tasks it was programmed to execute and there is no exchanging of tasks in response to an event.

Therefore, reconsideration is requested.

Respectfully submitted,

Date: May 29, 2007

Timothy N. Trop, Reg. No. 28,994
TROP, PRUNER & HU, P.C.
1616 South Voss Road, Suite 750
Houston, TX 77057-2631
713/468-8880 [Phone]
713/468-8883 [Fax]

Attorneys for Intel Corporation